

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

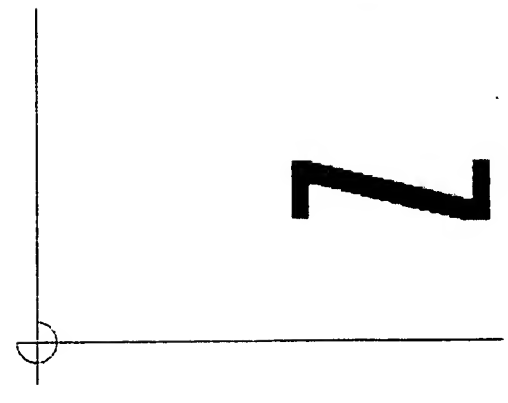
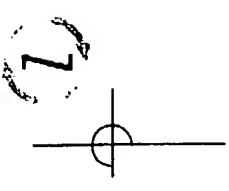
Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



# Zeiter a

January 2003

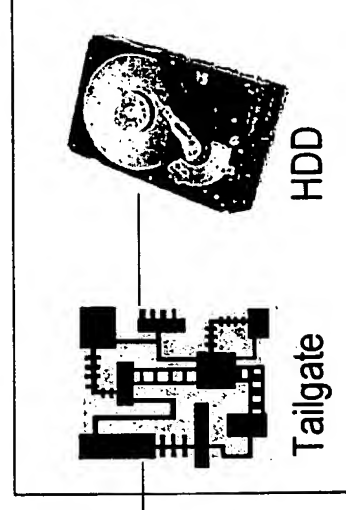
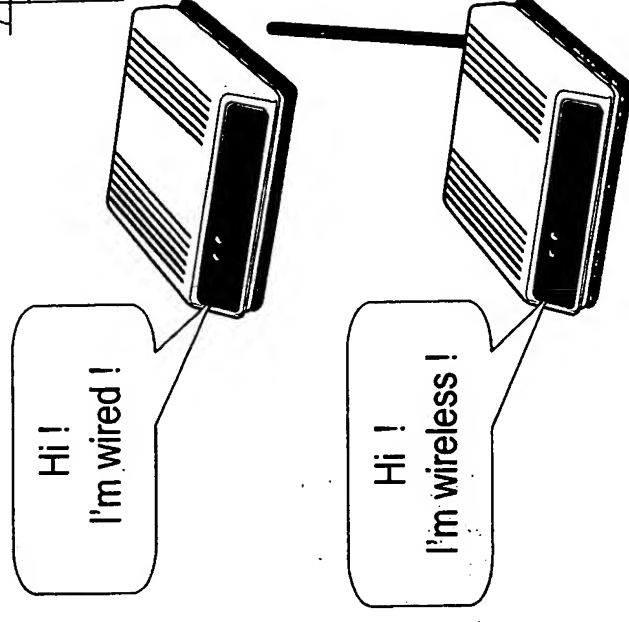
NO. 15  
363391

1/21/2003

CONFIDENTIAL

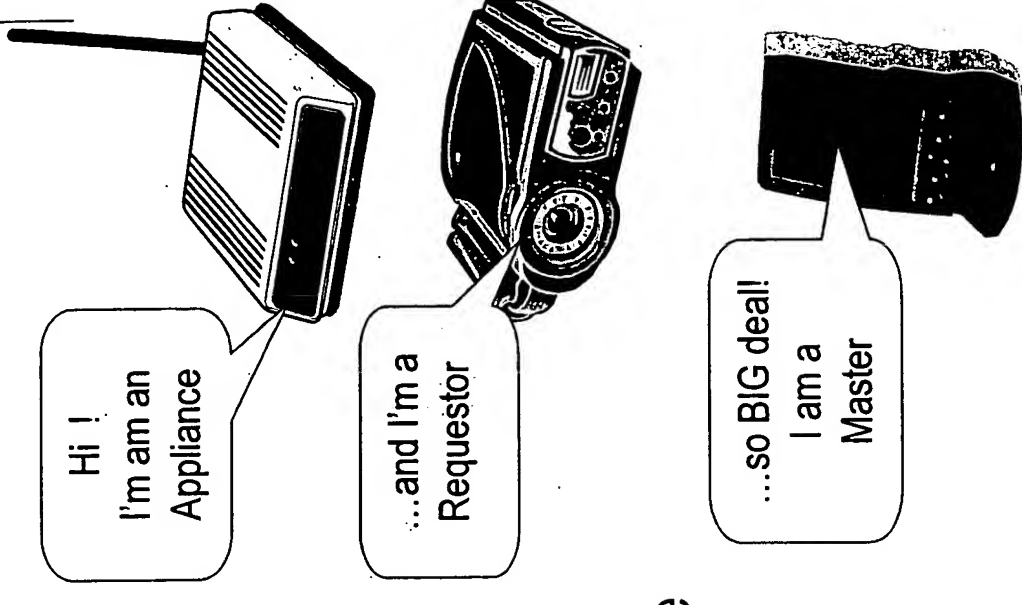
# What is a $\mu$ SAN ?

- A  $\mu$ SAN consists of a Storage Device, most likely an EIDE HDD, and a protocol adapter known as a "Tailgate" housed within a suitable enclosure
- The Tailgate adapter is very similar to existing 1394/USB adapters in complexity and cost
- The Tailgate provides the appropriate PHY and protocol level interface either wired or wireless determined by the needs of the product.



# μSAN - Some Basic Concepts

- A μSAN lives in an IP network
  - Ethernet
  - WiFi
  - HomePlug
  - HPnA
  - Any form of IP works
- A μSAN is a Peer on a network of Peers
- A μSAN is not a server – it's an appliance
- A μSAN can work under a master
- A μSAN user is not really a Client
- A μSAN user can be called a Requestor

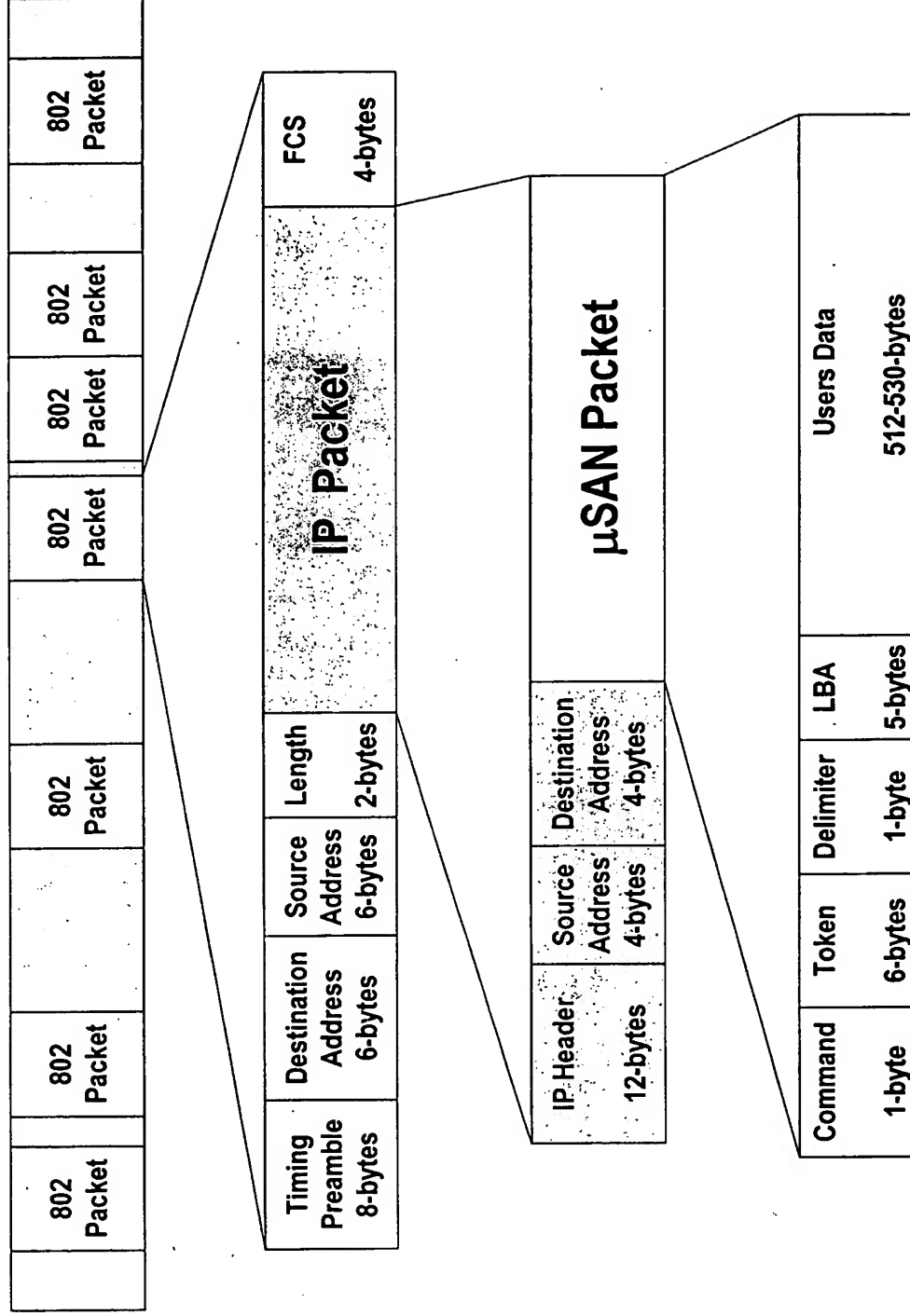


# How $\mu$ SAN Works

- Embedding the  $\mu$ SAN protocol inside IP
- $\mu$ SAN commands
- Discovering a networked  $\mu$ SAN device
- Owning your own storage volume
- Sharing data on a  $\mu$ SAN
- Security and authentication
- Spanning and redundancy
- Dissagregation - Real life examples

# Embedding the $\mu$ SAN protocol inside IP




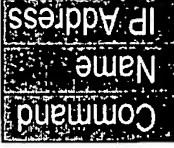
## Ethernet Stream



# $\mu$ SAN Broadcast IP Protocol

- $\mu$ SAN uses Broadcast IP Protocol in conjunction with UDP Transport Protocol to perform two functions:
  - Name Resolution
  - $\mu$ SAN Find (discovery)
- A “found” or discovered  $\mu$ SAN will identify the IP address of its Root to a requestor using a Unicast response directly to the Requestor.
- The Requestor must wait an appropriate length of time to ensure that all FIND RESPONSES are received and may want to retry to ensure none are lost.
- In true Peer to Peer fashion, the root has no name.

# μSAN Broadcast Protocol

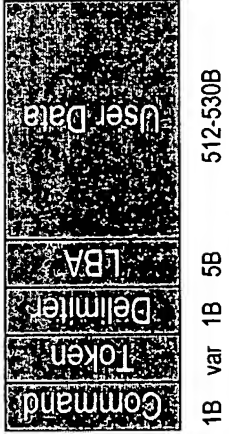
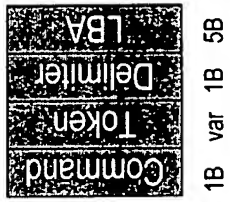


<b>μSAN Find Broadcast</b>	 1B	This request is issued by the Requester looking for the root IP of all μSAN on the LAN. Every μSAN that receives this packet must respond with μSAN Find Response packet to the calling IP/UDP Port. The μSAN Find packet is addressed to UDP Port XX
<b>μSAN Find Response</b>	 IP Address 1B 4b	Each μSAN in response to a μSAN Find packet responds to the Requester with its IP address in a μSAN Find Response packet. This packet is addressed to the requesting IP/UDP Port. It is understood that there may be many responses to the request issued on the network and will be arbitrated at the network layer
<b>Name Resolution Request</b>	 Name 1B 16B	The μSAN Requester issues this packet to UDP Port XX as a request for Name to IP resolution. The μSAN that recognizes this Name as one of its partitions responds with a Name Resolution Response Packet.
<b>Name Resolution Response</b>	 Name IP Address 1B 16B 4b	The μSAN issues this packet in response to the Name Resolution Request packet with the IP address associated with the Name. It is issued to the requesting IP/UDP Port. The name is resent in case of multiple requests and the "Sorcerer's Apprentice Syndrome" present in UDP



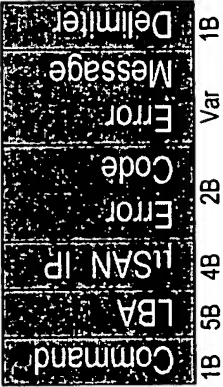
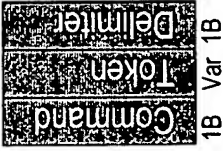
# Basic $\mu$ SAN Commands

- $\mu$ SAN Block Transfer Protocol is a master/slave architecture with the Requester as the master
- Operates under either UDP or TCP transport protocol and defaults to the most efficient mode
- Commands are variable length depending on function
- The block size is fixed at 512 (or 530) bytes
- LBA (Logical Block Address) field is 40 bits - 512 Tbytes

# Basic $\mu$ SAN Commands

<b>Transfer Command</b>		<p>This command is used to transfer the data either as a write to the <math>\mu</math>SAN or the result of a request from the <math>\mu</math>SAN. One block of data is transferred. In a write from Requester operation, this is the only command that is transferred from the Requester. The <math>\mu</math>SAN responds with an ACK Command. This command may be sent to either unicast or multicast destination IP addresses.</p>
<b>Request Command</b>		<p>This command is a request for a transfer from the <math>\mu</math>SAN. The <math>\mu</math>SAN responds with a Transfer Command containing the requested data. No ACK Command is required in response to this transfer. This command may be sent to either unicast or multicast destination IP addresses.</p>
<b>Request Lock Command</b>		<p>The function of this command is to provide for semaphore functionality of the data. This is a transfer request from the <math>\mu</math>SAN, however, it locks out all other Requestors. The lock is released when a Transfer Command is received from this Requester or a time-out occurs. This command may be sent to either unicast or multicast destination IP addresses</p>
<b>Ack Command</b>		<p>This command acknowledges a successful transfer. It is required if communication via UDP to acknowledge a successful transfer from a Requester to the <math>\mu</math>SAN. In TCP, a Requester may wait for an ACK to ensure single threaded error handling. The LBA that was received is returned avoiding the "Sorcerer's Apprentice" problem inherent in IP due to time-out retry. The <math>\mu</math>SAN IP address is the address of the <math>\mu</math>SAN that responded to the transfer.</p>

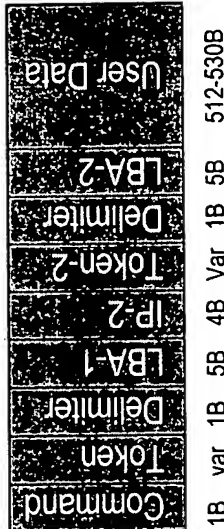
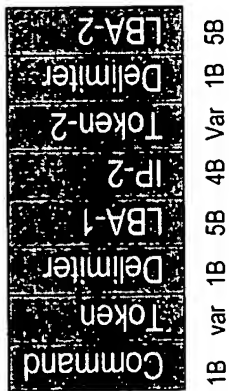
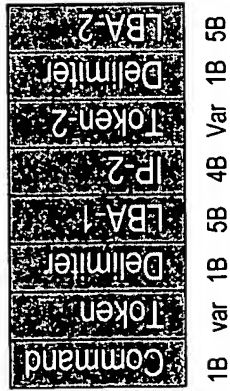
# Basic $\mu$ SAN Commands

<b>Error Command</b>		<p>The Error Command indicates that an operation could not be completed for some reason and is a transfer from the <math>\mu</math>SAN to the Requester. The <math>\mu</math>SAN IP address represents the address of the <math>\mu</math>SAN that responded to the transfer. (For example, a multicast GO TRANSFER command to a set of mirrored partitions may generate an ERROR from one of the <math>\mu</math>SAN back to the transferring <math>\mu</math>SAN who would then pass that ERROR back to the client for exception resolution. The ERROR that is passed back to the Requester will have the destination's <math>\mu</math>SAN IP address.)</p> <p><u>Error Codes (partial list):</u></p> <ul style="list-style-type: none"> <li>01h Invalid Authorization</li> <li>02h Partition has locked you out</li> <li>04h Go Command - Invalid Authorization at destination</li> <li>08h Go Command - locked Partition at the destination</li> <li>10h LBA is out of Range</li> <li>20h LBA is Write Protected</li> </ul>
<b>Release Partition Command</b>		<p>This command will cause the <math>\mu</math>SAN to release a partition, erase the data blocks within that partition, and release the IP/Name. The released blocks are placed back into the root pool for future allocation.</p>

# μSAN Extended- “Go” Commands

- Specifically used to copy a logical block from one μSAN partition to another μSAN partition.
- Initiated by the Requester who resolves both of the partition's Names to their relative IP address and passes the command to one of the μSANs.
- The result is to double the effective bandwidth and to remove a burden from the Requester.
- “Go” commands are not necessary for basic μSAN communication and are considered optional

# μSAN Extended- "Go" Commands

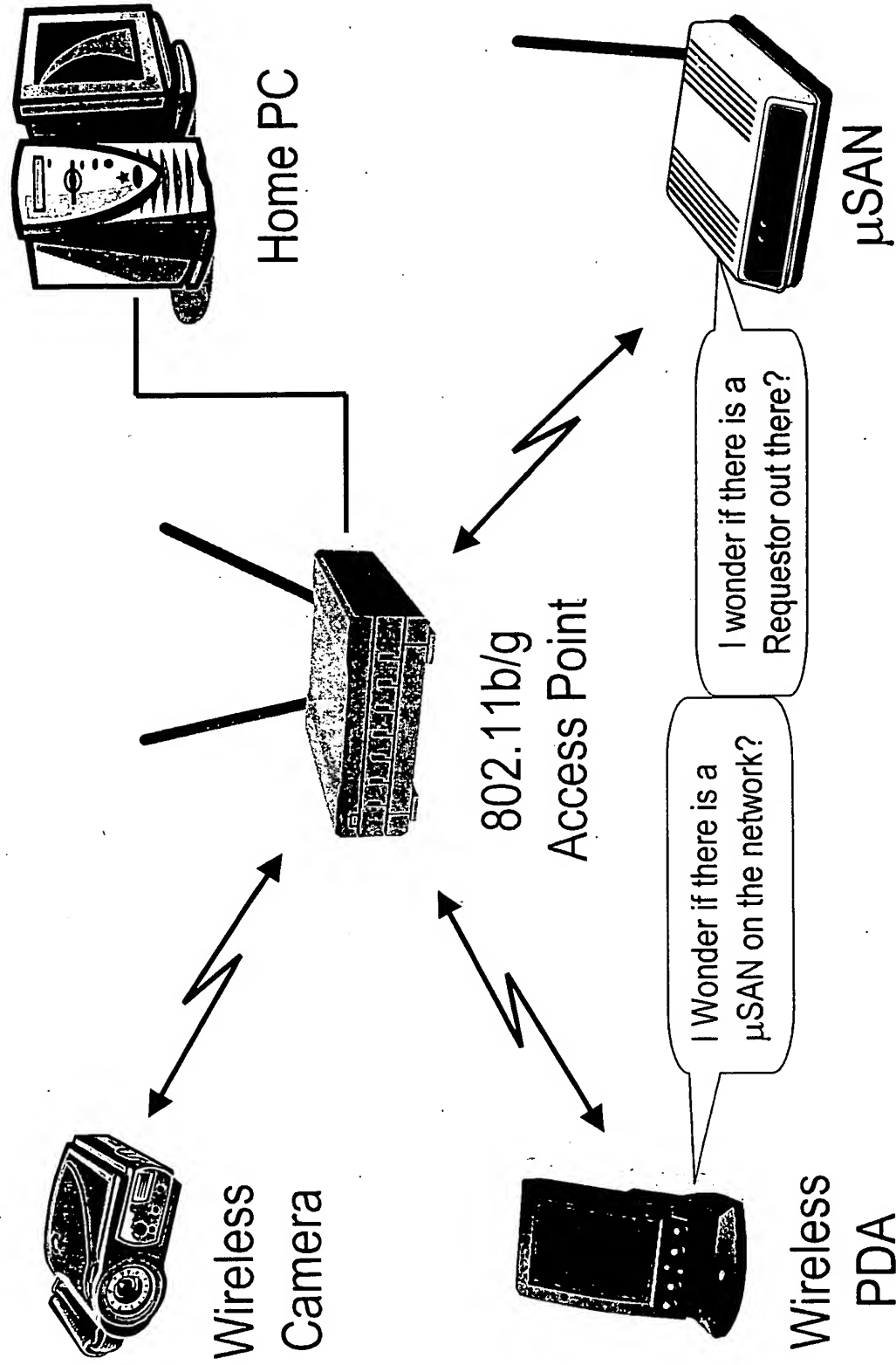
<b>Go Transfer Command</b>	 <p>1B var 1B 5B 4B Var 1B 5B 512-530B</p>	The Go Transfer Command causes an addressed μSAN to go active on the network and autonomously transfer a block of data to another μSAN identified in "IP-2"
<b>Go Request Command</b>	 <p>1B var 1B 5B 4B Var 1B 5B</p>	The Go Request Command is similar to the Go Transfer Command. The Requester issues this command to a μSAN to have it request data from another μSAN. At the completion of the transfer, the μSAN will ACK or ERROR the Requester. TCP transport is used to send the request to the second μSAN.
<b>Go Request Lock Command</b>	 <p>1B var 1B 5B 4B Var 1B 5B</p>	The Go Request Lock Command is similar to the Request Lock command in the basic command set except that it requests an addressed μSAN to lock access to another μSAN identified in "IP-2"

# μSAN Commands with Multi-Cast IP

- Optional command set specifically used to support Spanning and Mirroring among multiple μSANs.
- Requires Requestors as well as μSANs and routers to support Internet Group Management Protocol (IGMP).
- The burden on Requestors and μSANs to support IGMP is light, but support on routers is much heavier.
- These commands allow the μSAN to exploit the unique capabilities of Multi-Cast to provide Spanning and Mirroring.

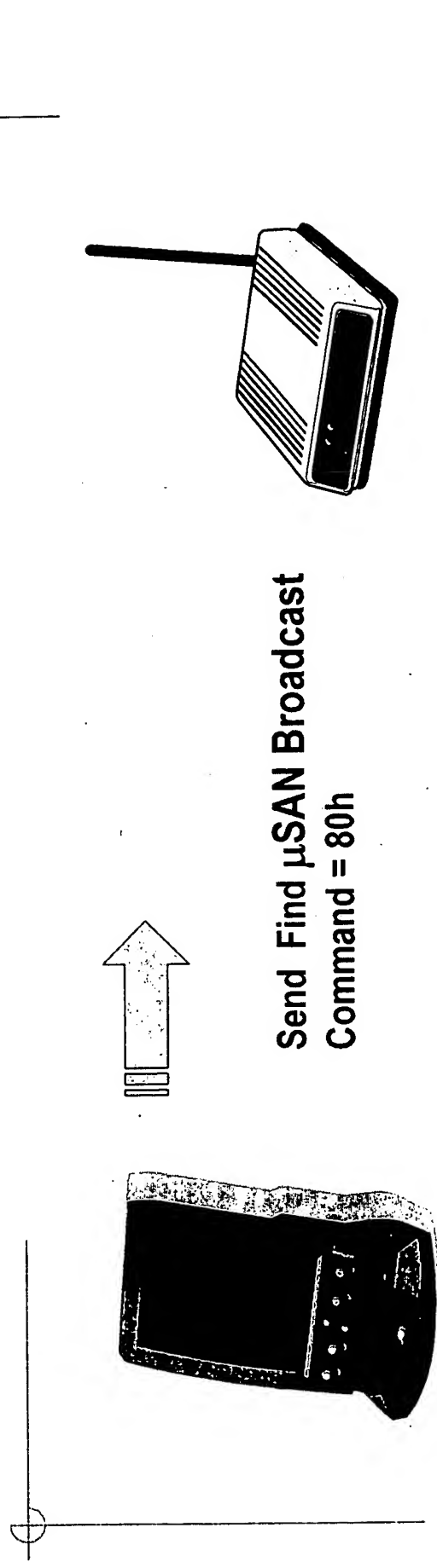
②

# Discovering a networked $\mu$ SAN device





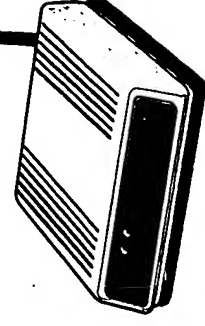
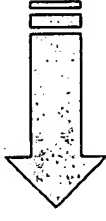
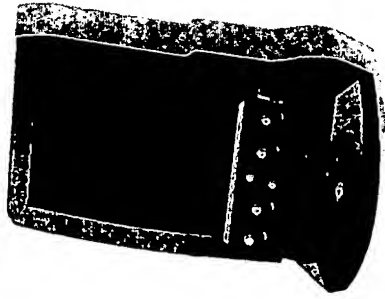
# Discovering a networked $\mu$ SAN device



I'm Looking for a  $\mu$ SAN

I'm Looking for a  $\mu$ SAN

# Discovering a networked $\mu$ SAN device

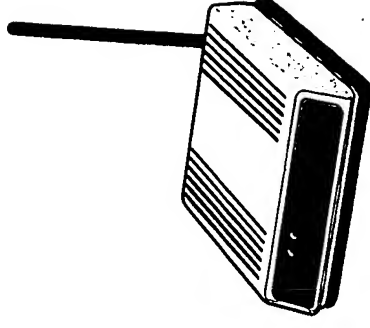
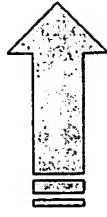
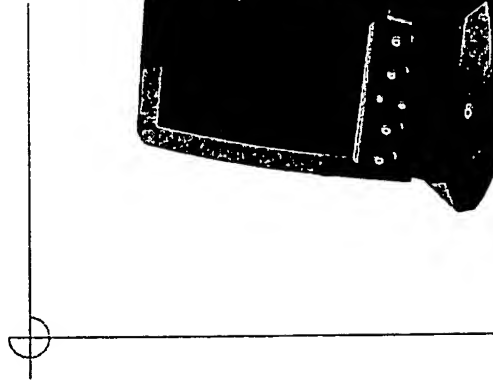


Send  $\mu$ SAN Find Response  
Command = 81h  
IP Addr = 192.29.100.0

$\mu$ SAN 192.29.100.0

I Hear you  
I'm here

# Owning a $\mu$ SAN storage volume

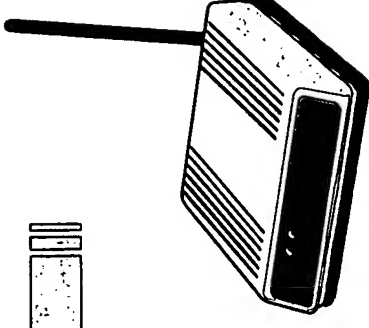
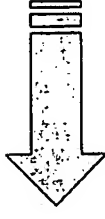
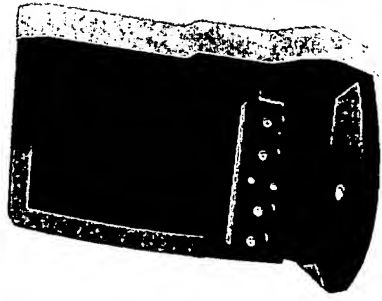


Get  $\mu$ SAN Root Status  
(Request Transfer Command)  
LBA = 0  
Token = (don't care)

$\mu$ SAN 192.29.100.0

How much Free Space  
is available ?

# Owning a $\mu$ SAN storage volume

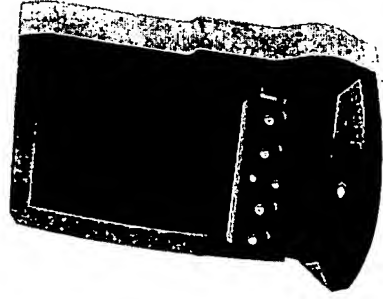


- a. Version
- b. Total Capacity
- c. Available Capacity
- d. Speed
- e. Reliability
- f. Portability
- g. QoS Capability
- h. ...

$\mu$ SAN 192.29.100.0

I have lots of space  
Here's my root status

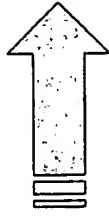
# Owning a $\mu$ SAN storage volume



$\mu$ SAN 192.29.100.0

OK!

Reserve a partition



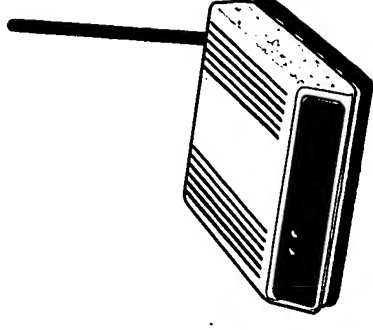
Allocate a Partition  
(Block Transfer Command)

LBA = 0

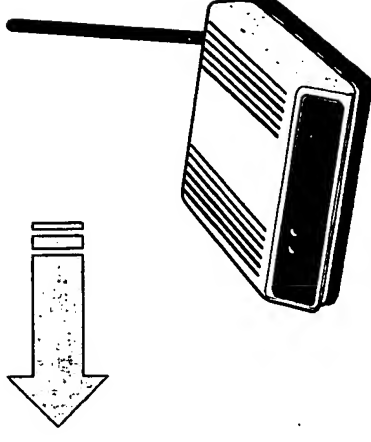
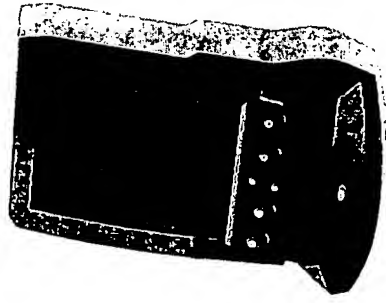
Token = (don't care)

Data bytes 0 – 511

- a. Partition Name = PDA
- b. Partition Token = PDA's token
- c. ID Character String
- d. Authentication Tags
- e. Partition Size
- f. Personality Tags



# Owning a $\mu$ SAN storage volume



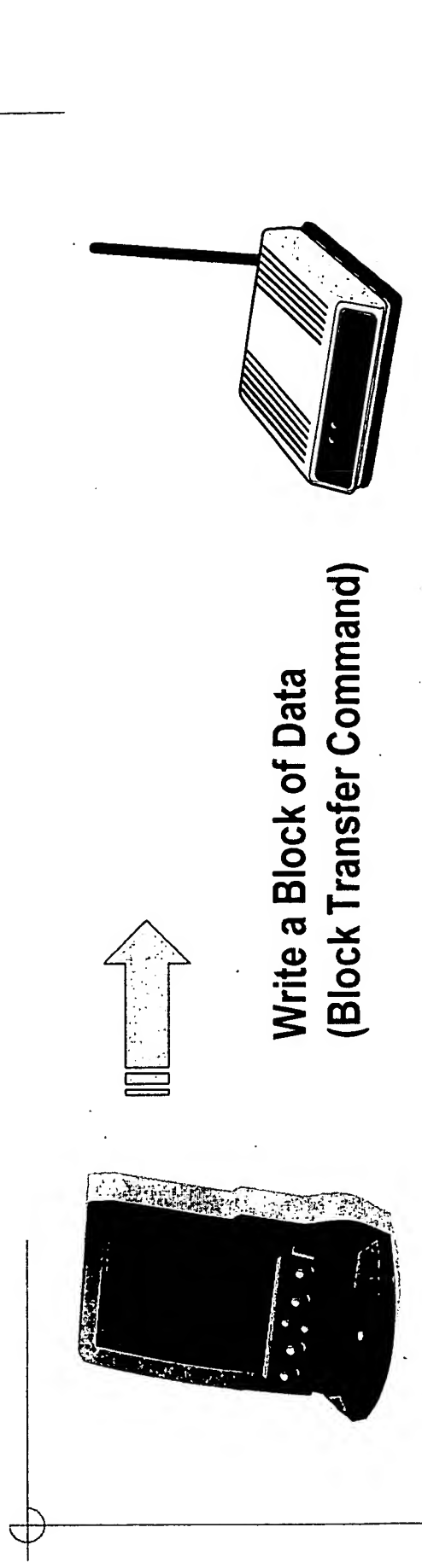
ACK

$\mu$ SAN 192.29.100.0

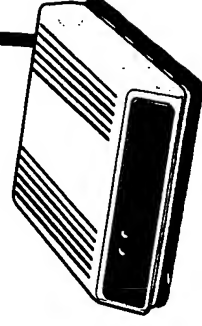
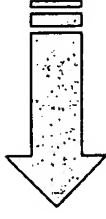
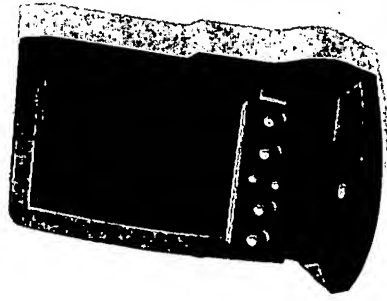
OK!

PDA's Partition is  
Reserved

# Owning a $\mu$ SAN storage volume



# Owning a $\mu$ SAN storage volume



ACK

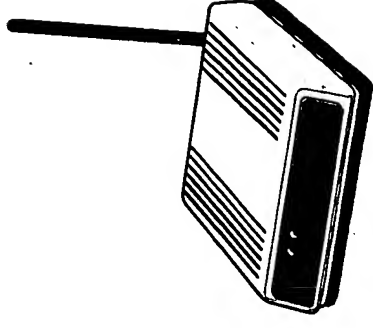
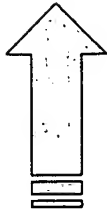
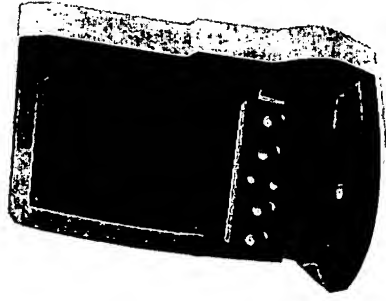
PDA Partition

OK

Data Block has been  
written



# Owning a $\mu$ SAN storage volume



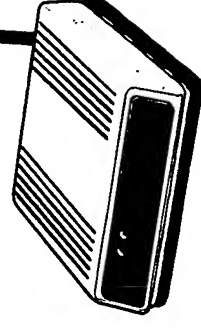
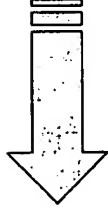
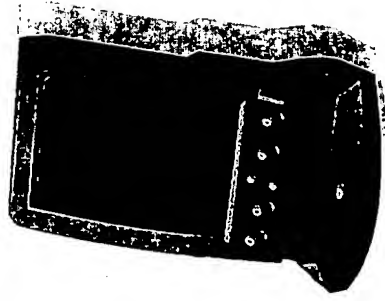
Read a block of data  
(Request Transfer Command)

LBA = 1000

Token = PDA's token

Hey – PDA Partition!  
I want to read back my  
data to verify it

# Owning a $\mu$ SAN storage volume



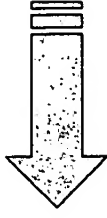
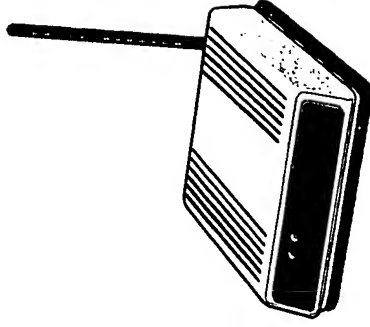
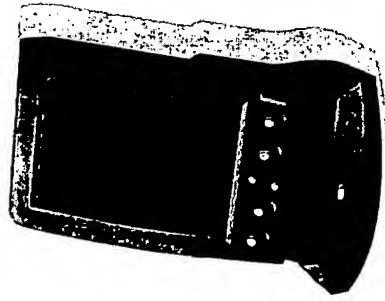
Block Data Transfer  
Block Address = 1000  
Data Bytes 0-511  
(PDA's Data)

PDA Partition

OK!

Here is your Data

# Sharing data on a $\mu$ SAN



Hey! Camera

I want to see the  
pictures you've taken

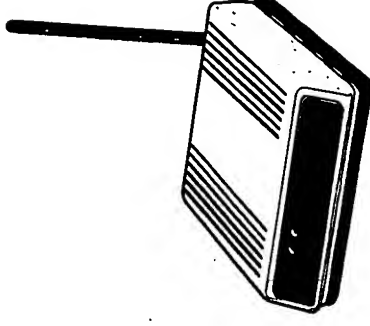
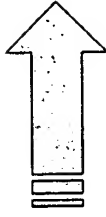
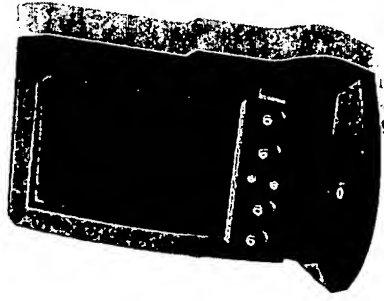
Hi PDA !

I save them in  $\mu$ SAN  
192.29.100.0

Here's my token

And Partition Name

# Owning a $\mu$ SAN storage volume



Read a Block of Data  
(Request Transfer Command)

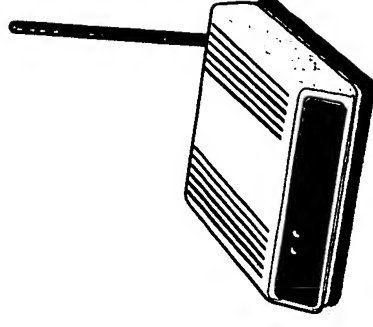
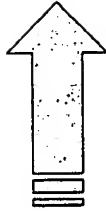
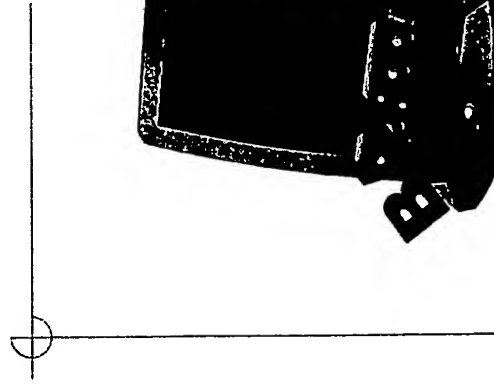
LBA = 1000

Token = Camera's Token

Hey Camera Partition!

I Want to see pictures

# Security and authentication



**Hey PDA Partition!**

**I Want to erase data.**

**(chuckle)**

**Write Data Block**

**(Block Transfer Cmd)**

**Requestor IP = Wrong**

**Requestor MAC = Wrong**

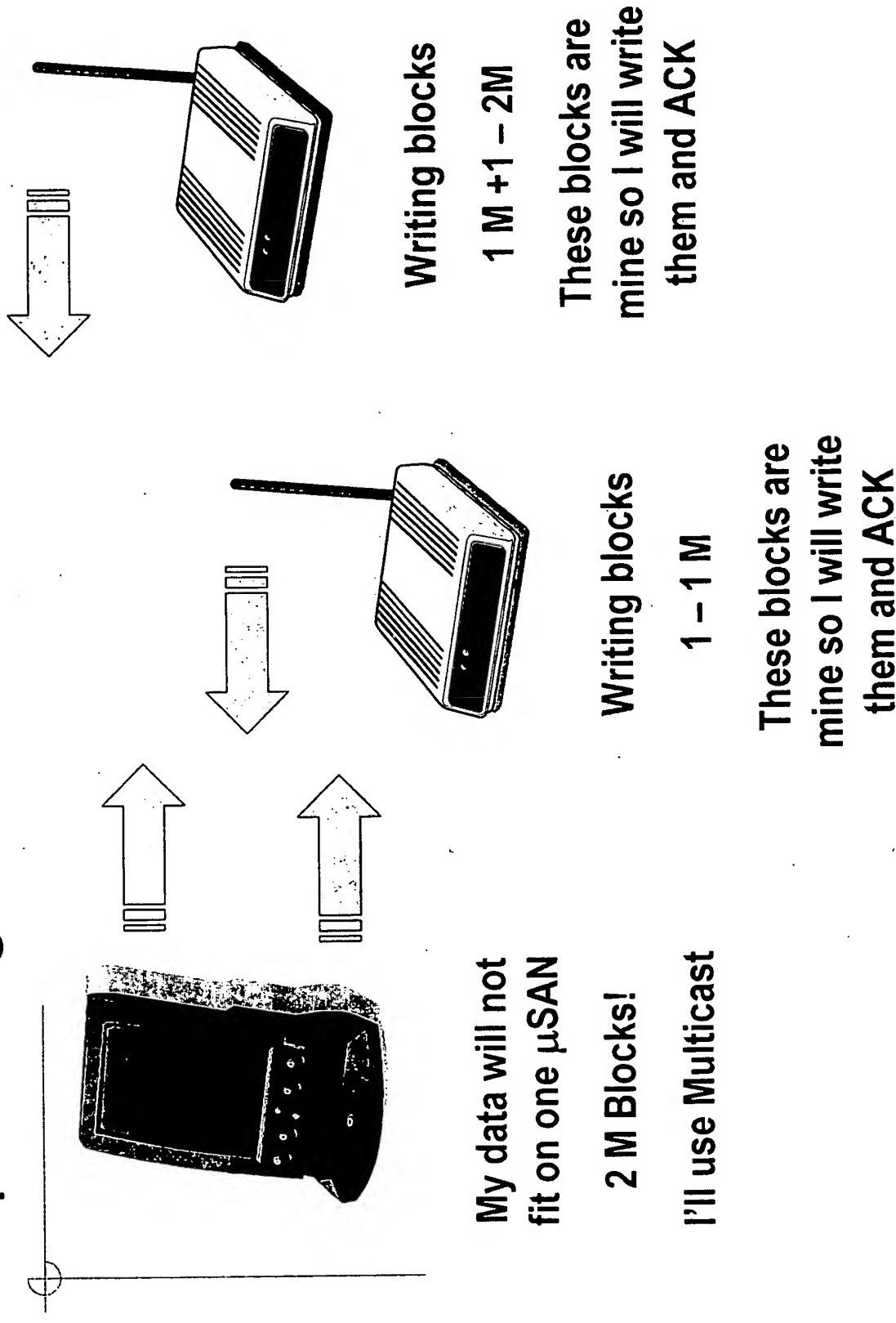
**LBA DAC = Wrong**

**Payload DAC = Bogus**

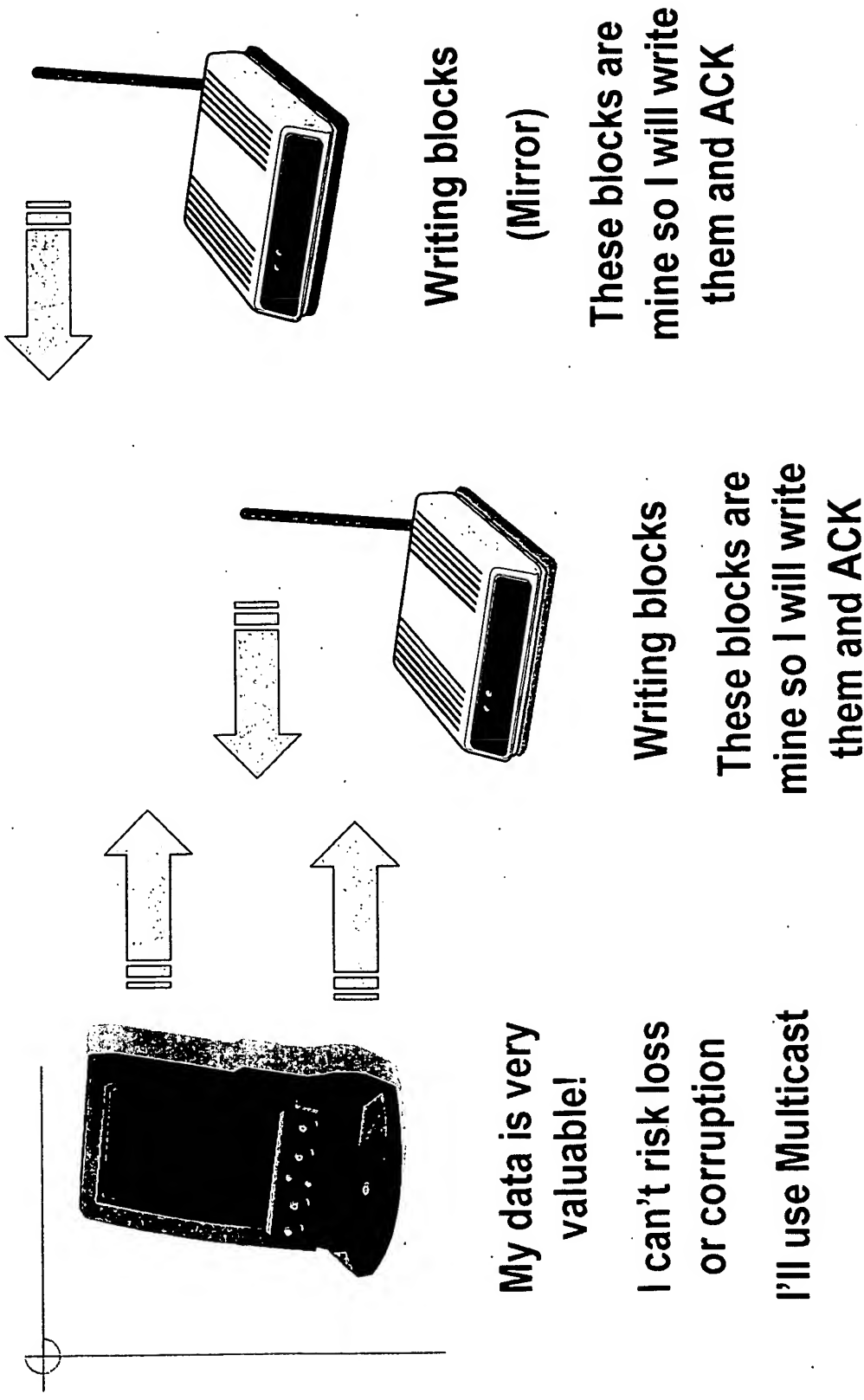
**Not even close!**

**No Response**

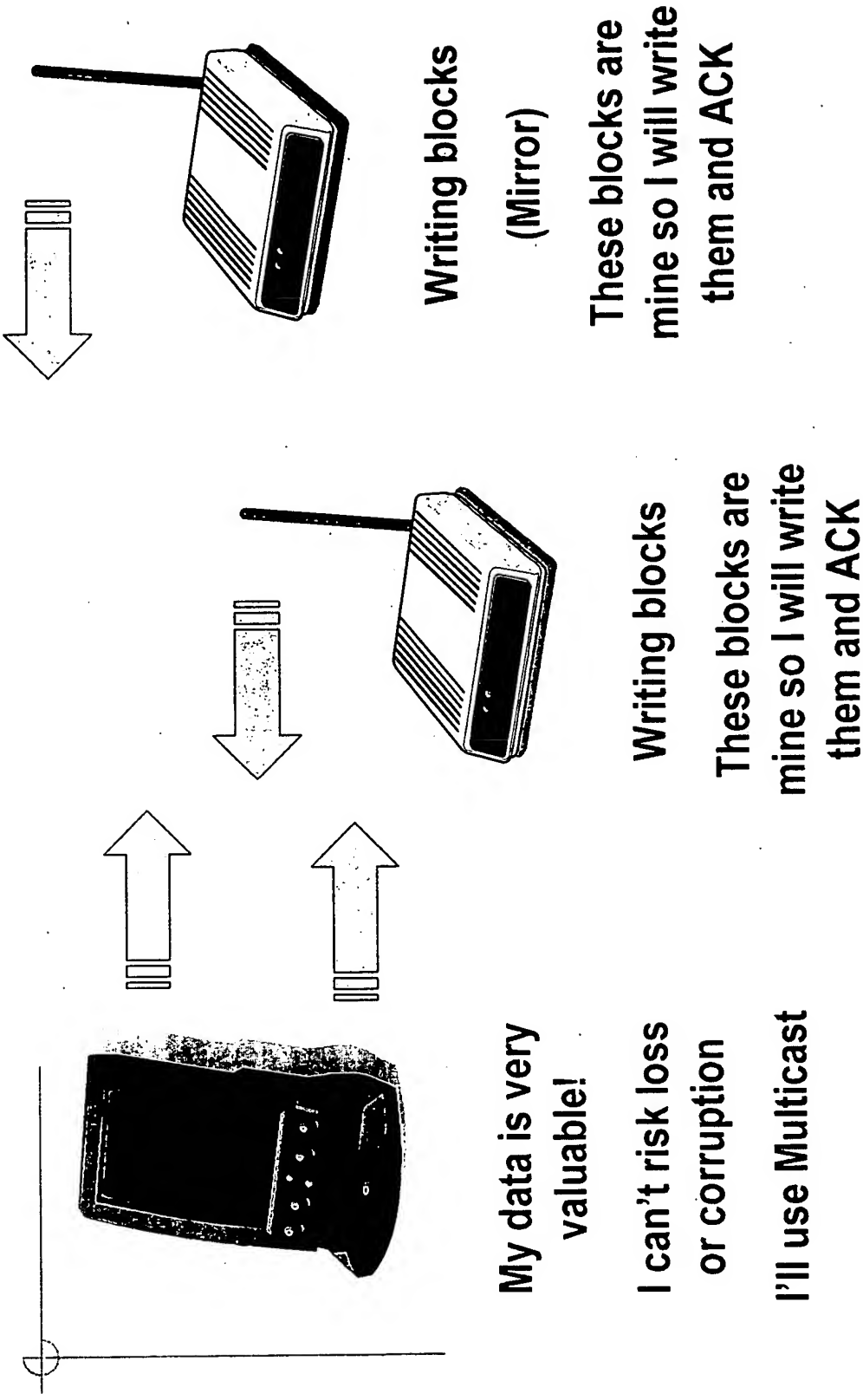
# Spanning Partitions with Multicast



# Redundant Partitions with Multicast

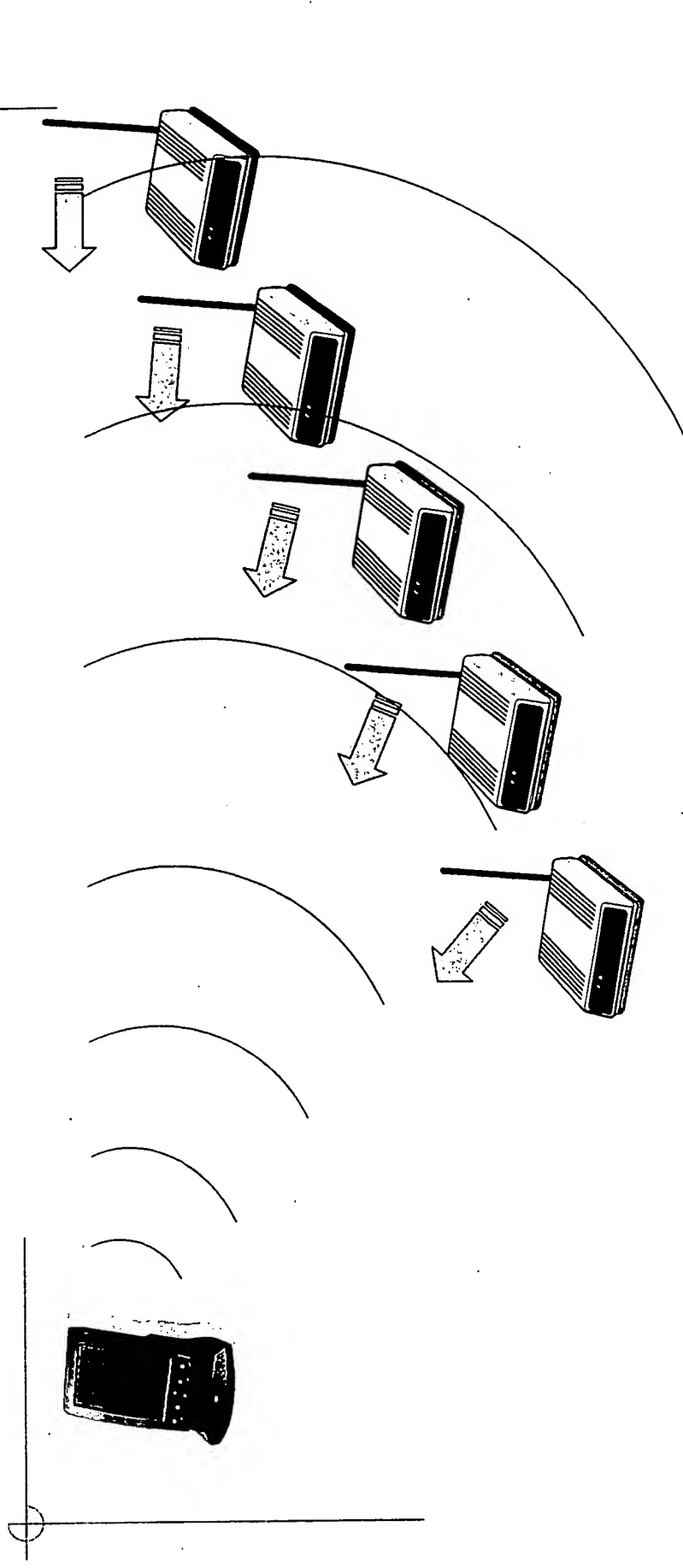


# Striping Data with Multicast

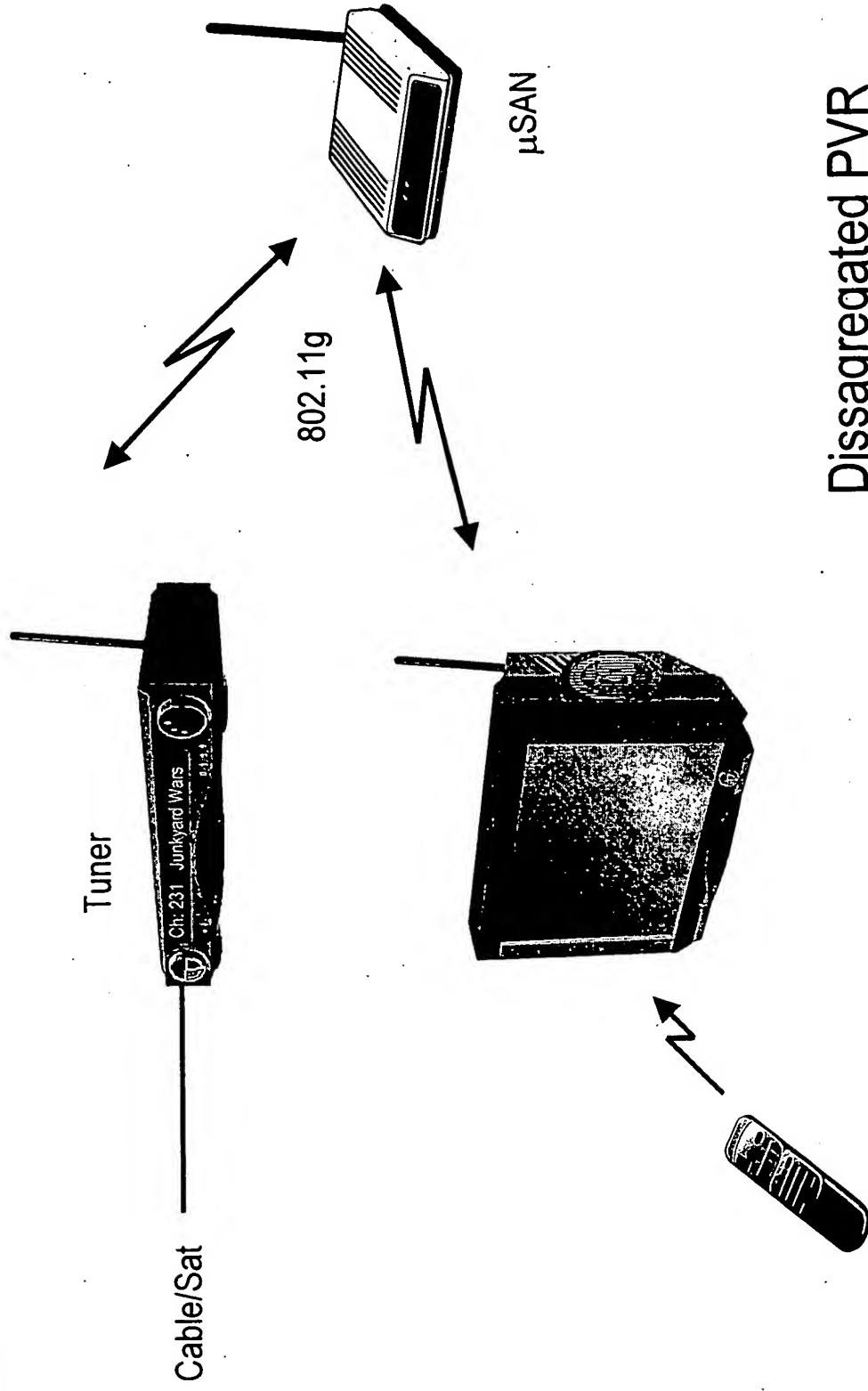




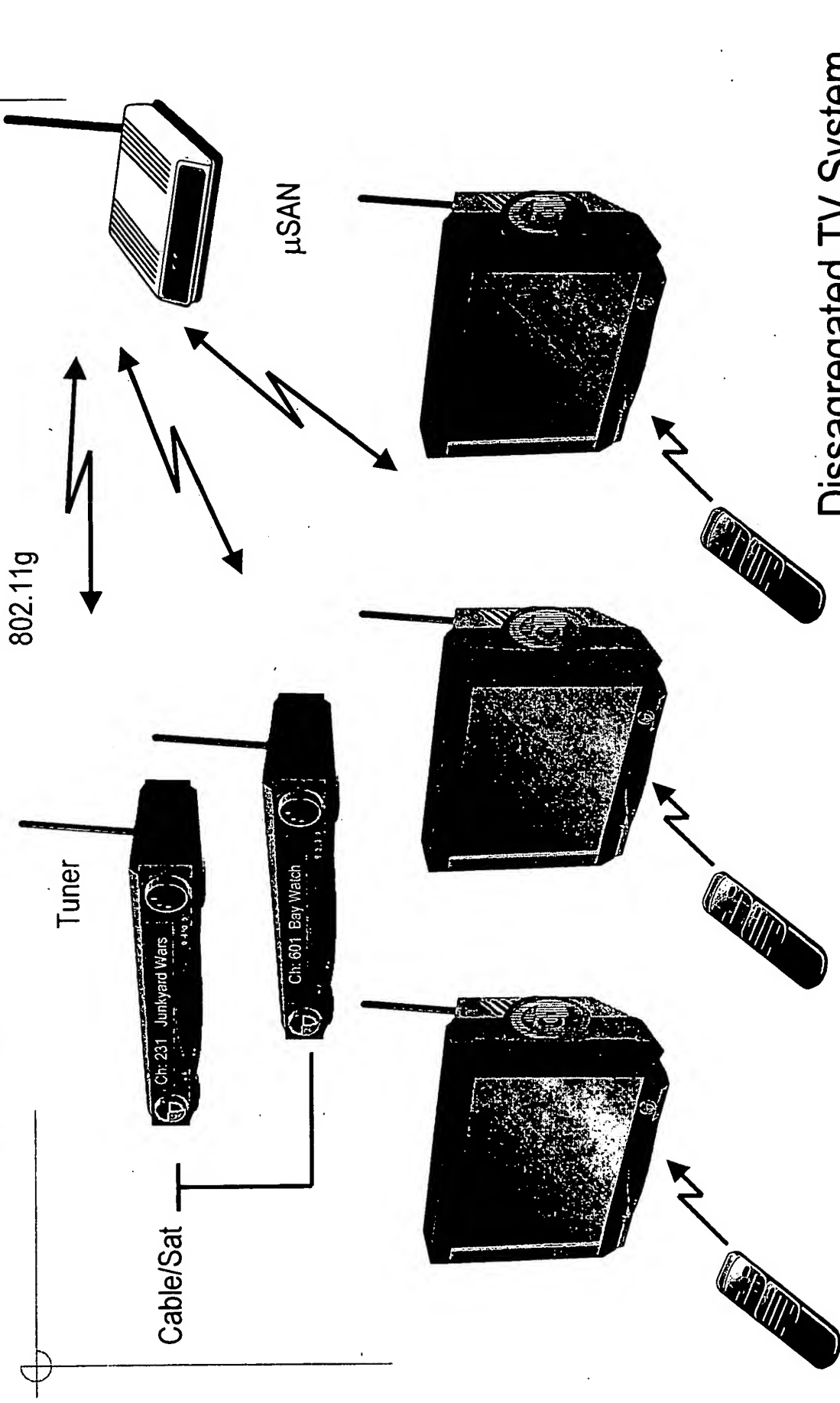
# RAID with Multicast



# Dissagregation - Real life examples



# Dissagregation - Real life examples



Dissagregated TV System

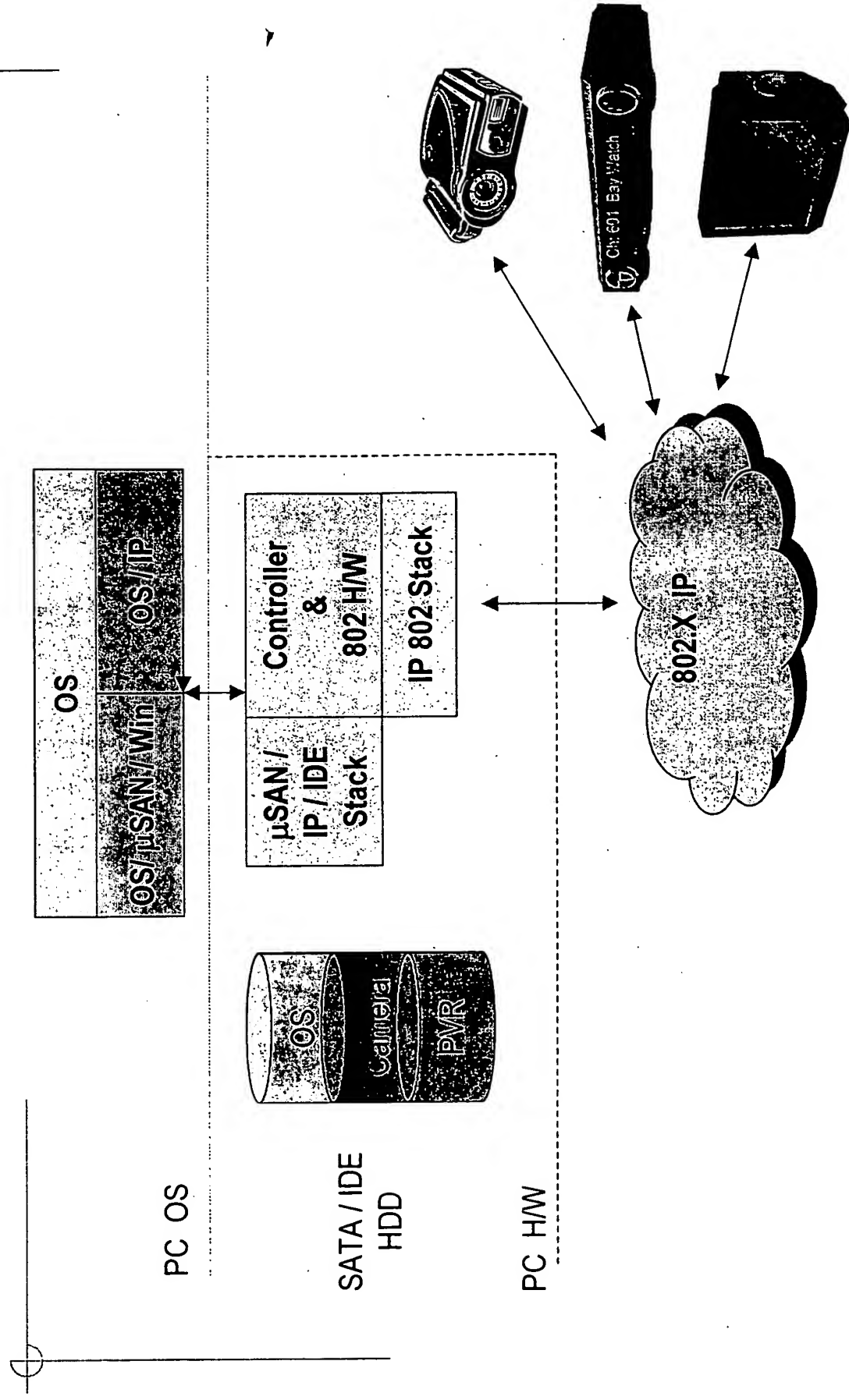
# Competitive Technologies – IP

	<b>µSAN</b>	<b>NAS</b>	<b>iSCSI</b>
<b>Connection</b>	Internetwork Protocol	Internetwork Protocol	Internetwork Protocol
<b>Data Access Method</b>	Block Level	File Level	Block Level
<b>File Format</b>	None imposed. Client defines	Industry standards i.e. FAT16, FAT32, NTFS, HPFS, Unix, Solaris ...	None imposed. Client defines
<b>O/S Support</b>	Agnostic – all supported	Support defined by OS compatibility options	Agnostic – all supported
<b>Master requirement</b>	None required	NAS is a master	Master required on each subnet
<b>Control Processor</b>	32 bit Low-Cost Embedded RISC	Pentium Class or greater	Hi-performance RISC with H/W acceleration (TOEs)
<b>DHCP Requirement</b>	None for simple networks. DHCP on local net or internal for complex networks	Contains DHCP services	Relies on Master

# Competitive Technologies - IP (con't)

	<b>μSAN</b>	<b>NAS</b>	<b>iSCSI</b>
Multiple Volumes (Partitions)	Real Volumes – Created, used, shared and deleted by client with ownership rights	Supported as volumes or extensions defined by the OS	Virtualized Volumes – Supported through <u>master</u> with ownership rights
Volume Addressing	Volumes are addressed by a unique IP address	Virtual Volume support only – Directroy structured	Partitions are accessed by a master controller using a directory lookup service
Number of volumes	No fixed limit	No fixed limit	No fixed limit
Discovery	Supports peer discovery, UPnP and DNS depending upon environment	OS dependent – NetBIOS, DNS, Master Brower	iSNS – iSCSI master Name Server
Authentication	Supports light, efficient and secure rotating key authentication across both private IP networks and through the Internet	OS supports many forms of authentication including Kerberos, PKI, IPsec, etc. on both private IP networks and through the Internet	Relies on heavy IPsec security services from connected masters on private IP networks
Controller cost	\$10 - \$15	\$100+	\$500+

# Embedded ubiquitous $\mu$ SAN



## **EXHIBIT A – US 60/441739**